This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Currently Amended) A method for processing an input file in a file system, wherein the input file has an input file name, comprising:

providing a data structure generated by applying a function to all file names in a file system to determine values corresponding to the file names, wherein the data structure indicates those values corresponding to the file names to indicate all file names used in the file system;

applying a function to map the input file name to a value; and

processing the data structure to determine whether there is a preexisting file in the file system having a name that maps, according to the function, to the same value to which the input file name maps, wherein two files that map to a same value according to the function are capable of having a same name.

2. (Original) The method of claim 1, wherein the mapped-to values require fewer bits of storage than the file names.

3. (Original) The method of claim 1, wherein the function is a hash function that maps the input file name to an integer value, and wherein the data structure includes an entry for each possible integer value capable of being generated from the hash function.

4. (Original) The method of claim 3, wherein processing the data structure to determine whether there is a preexisting file comprises determining whether the entry for the integer value to which the input file name maps indicates the presence of one preexisting file mapping to the same integer value as the input file name.

5. (Original) The method of claim 4, wherein the data structure is a one-dimensional array and wherein each entry is capable of having one of two values, further comprising setting the entry to a first value if there is one preexisting file name in the file system that maps to the integer value for the entry, and wherein determining whether there is one preexisting file comprises determining whether the entry for the integer value to which the input file name maps has the first value.

6. (Original) The method of claim 1, further comprising:

applying the function to each file name in the file system to map each file name to one value; and

indicating in the data structure, for each file name, that there is one preexisting file for the value to which the file name maps.

7. (Original) The method of claim 6, wherein the input file is the subject of an access request, further comprising scanning each file in the file system to determine if there is at least one preexisting file having the same name as the input file name if there is one preexisting file in the file system having a name that maps, according to the function, to the same value to which the input file name maps.

8. (Original) The method of claim 7, wherein the access request to the input file is to add the input file as a new file to the file system, further comprising:

adding the input file as a new file to the file system if no preexisting file in the file system has the same name as the input file name; and

rejecting the access request if there is a preexisting file in the file system having the same name.

9. (Original) The method of claim 7, wherein the access request to the input file is to update a file in the file system with data from the input file, further comprising:

updating a preexisting file in the file system having the same name as the input file with the data in the input file if there is such a preexisting file; and

rejecting the access request if there is no preexisting file in the file system having the same name as the input file name.

10. (Currently Amended) A system for processing an input file in a file system, wherein the input file has an input file name, comprising:

means for providing a data structure generated by applying a function to all file names in a file system to determine values corresponding to the file names, wherein the data structure indicates those values corresponding to the file names to indicate all file names used in the file system;

means for applying a function to map the input file name to a value; and

means for processing the data structure to determine whether there is a preexisting file in the file system having a name that maps, according to the function, to the same value to which the input file name maps, wherein two files that map to a same value according to the function are capable of having a same name.

11. (Original) The system of claim 10, wherein the mapped-to values require fewer bits of storage than the file names.

12. (Original) The system of claim 10, wherein the function is a hash function that maps the input file name to an integer value, and wherein the data structure includes an entry for each possible integer value capable of being generated from the hash function.

13. (Original) The system of claim 10, wherein the means for processing the data structure to determine whether there is a preexisting file comprises determining whether the entry for the integer value to which the input file name maps indicates the presence of one preexisting file mapping to the same integer value as the input file name.

14. (Original) The system of claim 13, wherein the data structure is a one-dimensional array and wherein each entry is capable of having one of two values, further comprising setting the entry to a first value if there is one preexisting file name in the file system that maps to the integer value for the entry, and wherein determining whether there is one preexisting file comprises determining whether the entry for the integer value to which the input file name maps has the first value.

15. (Original) The system of claim 10, further comprising:

means for applying the function to each file name in the file system to map each file name to one value; and

means for indicating in the data structure, for each file name, that there is one preexisting file for the value to which the file name maps.

16. (Original) The system of claim 10, wherein the input file is the subject of an access request, further comprising means for scanning each file in the file system to determine if there is at least one preexisting file having the same name as the input file name if there is one preexisting file in the file system having a name that maps, according to the function, to the same value to which the input file name maps.

17. (Original) The system of claim 16, wherein the access request to the input file is to add the input file as a new file to the file system, further comprising:

means for adding the input file as a new file to the file system if no preexisting file in the file system has the same name as the input file name; and

means for rejecting the access request if there is a preexisting file in the file system having the same name.

18. (Original) The system of claim 16, wherein the access request to the input file is to update a file in the file system with data from the input file, further comprising:

means for updating a preexisting file in the file system having the same name as the input file with the data in the input file if there is such a preexisting file; and

means for rejecting the access request if there is no preexisting file in the file system having the same name as the input file name.

19. (Amended) An article of manufacture for processing an input file in a file system, wherein the input file has an input file name, the article of manufacture comprising computer usable media including at least one computer program embedded therein that causes the computer to perform:

providing a data structure generated by applying a function to all file names in a file system to determine values corresponding to the file names, wherein the data structure indicates those values corresponding to the file names to indicate all file names used in the file system;

applying a function to map the input file name to a value; and

processing the data structure to determine whether there is a preexisting file in the file system having a name that maps, according to the function, to the same value to which the input file name maps, wherein two files that map to a same value according to the function are capable of having a same name.

20. (Original) The article of manufacture of claim 19, wherein the mapped-to values require fewer bits of storage than the file names.

21. (Original) The article of manufacture of claim 19, wherein the function is a hash function that maps the input file name to an integer value, and wherein the data structure includes an entry for each possible integer value capable of being generated from the hash function.

22. (Original) The article of manufacture of claim 21, wherein processing the data structure to determine whether there is a preexisting file comprises determining whether the entry

for the integer value to which the input file name maps indicates the presence of one preexisting file mapping to the same integer value as the input file name.

23. (Original) The article of manufacture of claim 22, wherein the data structure is a one-dimensional array and wherein each entry is capable of having one of two values, further comprising setting the entry to a first value if there is one preexisting file name in the file system that maps to the integer value for the entry, and wherein determining whether there is one preexisting file comprises determining whether the entry for the integer value to which the input file name maps has the first value.

24. (Original) The article of manufacture of claim 19, further comprising:

applying the function to each file name in the file system to map each file name to one value; and

indicating in the data structure, for each file name, that there is one preexisting file for the value to which the file name maps.

25. (Original) The article of manufacture of claim 24, wherein the input file is the subject of an access request, further comprising scanning each file in the file system to determine if there is at least one preexisting file having the same name as the input file name if there is one preexisting file in the file system having a name that maps, according to the function, to the same value to which the input file name maps.

26. (Original) The article of manufacture of claim 25, wherein the access request to the input file is to add the input file as a new file to the file system, further comprising:

adding the input file as a new file to the file system if no preexisting file in the file system has the same name as the input file name; and

rejecting the access request if there is a preexisting file in the file system having the same name.

27. (Original) The article of manufacture of claim 25, wherein the access request to the input file is to update a file in the file system with data from the input file, further comprising:

updating a preexisting file in the file system having the same name as the input file with the data in the input file if there is such a preexisting file; and

rejecting the access request if there is no preexisting file in the file system having the same name as the input file name.

28. (Previously Added) The method of claim 1, further comprising:

searching the file system for one preexisting file having the same name as the input file name if the data structure indicates that one preexisting file has a name that maps, according to the function, to the same value to which the input file maps; and

performing an operation if the file system includes one preexisting file having the same name as the input file.

29. (Previously Added) The method of claim 28, wherein performing the operation comprises:

applying update data to the preexisting file having the same name as the input file if the file system includes one preexisting file having the same name as the input file.

30. (Previously Added) The method of claim 28, wherein the input file comprises a file to add to the file system, further comprising:

returning an error if the file system includes one preexisting file having the same name as the input file; and

adding the input file to the file system if the file system does not include one preexisting file having the same name as the input file.

31. (Previously Added) The system method of claim 1, further comprising:

means for searching the file system for one preexisting file having the same name as the input file name if the data structure indicates that one preexisting file has a name that maps, according to the function, to the same value to which the input file maps; and

means for performing an operation if the file system includes one preexisting file having the same name as the input file.

32. (Previously Added) The system of claim 31, wherein the means for performing the operation further performs:

applying update data to the preexisting file having the same name as the input file if the file system includes one preexisting file having the same name as the input file.

33. (Previously Added) The system of claim 31, wherein the input file comprises a file to add to the file system, further comprising:

means for returning an error if the file system includes one preexisting file having the same name as the input file; and

means for adding the input file to the file system if the file system does not include one preexisting file having the same name as the input file.

34. (Previously Added) The article of manufacture of claim 19, further comprising:

searching the file system for one preexisting file having the same name as the input file name if the data structure indicates that one preexisting file has a name that maps, according to the function, to the same value to which the input file maps; and

performing an operation if the file system includes one preexisting file having the same name as the input file.

35. (Previously Added) The article of manufacture of claim 34, wherein performing the operation comprises:

applying update data to the preexisting file having the same name as the input file if the file system includes one preexisting file having the same name as the input file.

36. (Previously Added) The article of manufacture of claim 34, wherein the input file comprises a file to add to the file system, further comprising:

returning an error if the file system includes one preexisting file having the same name as the input file; and

adding the input file to the file system if the file system does not include one preexisting file having the same name as the input file.

37. (Previously Added) The method of claim 1, wherein the function comprises a wide hash function to produce a large number of possible hash values to minimize the likelihood that the application of the hash function to file names in the file system would have a same hash value.

38. (Previously Added) The system of claim 10, wherein the function comprises a wide hash function to produce a large number of possible hash values to minimize the likelihood that the application of the hash function to file names in the file system would have a same hash value.

39. (Previously Added) The article of manufacture of claim 19, wherein the function comprises a wide hash function to produce a large number of possible hash values to minimize the likelihood that the application of the hash function to file names in the file system would have a same hash value.

40. (New) The method of claim 1, further comprising:

storing the data structure indicating all file names used in the file system in cache memory, wherein the data structure in cache memory is scanned to determine whether there is a preexisting file without reading directory information from storage locations in a storage device.

41. (New) The method of claim 1, wherein the data structure includes multiple columns for different directories in the file system to indicate file names in different directories of the file system.

42. (New) The system of claim 10, further comprising:

means for storing the data structure indicating all file names used in the file system in cache memory, wherein the data structure in cache memory is scanned to determine whether there is a preexisting file without reading directory information from storage locations in a storage device.

43. (New) The system of claim 10, wherein the data structure includes multiple columns for different directories in the file system to indicate file names in different directories of the file system.

44. (New) The article of manufacture of claim 19, further comprising:

storing the data structure indicating all file names used in the file system in cache memory, wherein the data structure in cache memory is scanned to determine whether there is a preexisting file without reading directory information from storage locations in a storage device.

45. (New) The article of manufacture of claim 19, wherein the data structure includes multiple columns for different directories in the file system to indicate file names in different directories of the file system.